

## **REMARKS**

[0001] Applicants wish to thank Examiners Hillary and Feild for the telephone interview on March 9, 2004. In that interview, the prior art reference Blackman (see below) was discussed in relation to claims 1 and 2. In particular, the claim phrases “Information Management System” and “IMS message definition” were discussed in relation to Blackman. This discussion was helpful in preparing this response.

[0002] Claims 1-30 are pending in the case. The Examiner objected to claims 3 and 13 for informalities. The Examiner rejected claims 4, 10, 14, 20, 24, and 30 under 35 U.S.C. §112, second paragraph as being indefinite.

[0003] The Examiner rejected claims 1-30 under 35 U.S.C. §103(a) as obvious in view of U.S. Patent No. 5,737,597 to Blackman et al. (hereinafter “Blackman”), a publication from W3C on WIDL (hereinafter “WIDL”), U.S. Patent No. 6,038,393 to Lyengar et al. (hereinafter “Lyengar”), a publication on “XMI Opens Application Interchange” by Brodsky (hereinafter “Brodsky”), and U.S. Patent No. 6,182,029 to Friedman et al. (hereinafter “Friedman”).

[0004] Applicant has amended claims 1-7, 9-17, 19-27, and 29-30 to address the objections and informalities indicated by the Examiner and to clarify the invention. The amended claims are believed to be in condition for allowance, and applicant respectfully requests the prompt allowance of claims 1-30.

### **REJECTION OF CLAIMS 4, 10, 14, 20, 24, AND 30 UNDER 35 U.S.C. §112, 2<sup>nd</sup> PARAGRAPH**

[0005] The Examiner rejected claims 4, 10, 14, 20, 24, and 30 under 35 USC §112, 2<sup>nd</sup> paragraph. Specifically, the Examiner rejected these claims based on use of the terms “relatively language independent” and “at least one...” Applicant has amended claims 4, 10, 14, 19, 20, 24, and 30 to more precisely describe and identify that which the Applicant considers his/her invention.

[0006] The term “relatively language independent” was replaced with language indicating that the Adata file includes a message definition substantially semantically equivalent to the message definition set forth in the original source code. The term “substantially” has been held

to be a definite term. *See* MPEP §2173.05(b). This amendment is supported in the specification on page 6, lines 13-17. The concept of semantic equivalence is also supported in the provisional patent application No. 60/151,479 filed 8/30/1999 on page 16 (a copy is included herewith for convenience). Applicants respectfully submit that amended claims 4, 10, 14, 19, 20, 24, and 30 are sufficiently definite as required under 35 USC §112, 2<sup>nd</sup> paragraph.

**REJECTION OF CLAIMS 1-5, 11-15, AND 21-25 UNDER 35 U.S.C. §103(a)**

[0007] The Examiner rejected claims 1-5, 11-15, and 21-25 in view of Blackman and WIDL. These rejections are respectfully traversed.

[0008] The Examiner bears the initial burden of establishing a *prima facie* case of obviousness. *See* MPEP § 2142, *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). The Examiner must show some objective teaching that suggests the claimed subject matter. *See In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). Furthermore, to establish a *prima facie* case of obviousness, the combination of the prior art references must teach or suggest all the claim limitations. *See* MPEP § 2142. And finally, any suggestion or motivation to combine references must be based on objective evidence of record.” *In re Lee*, 277 F.3d 1338, 61 USPQ2d 1430 (Fed. Cir. 2002).

[0009] Applicant respectfully asserts that the Examiner has failed to provide any objective teachings that suggest the claimed subject matter. Specifically, amended claim 1 recites in pertinent part “generating an XML document template from **a transaction processing system message definition...**” (emphasis added) The Examiner has provided no objective evidence in Blackman or the WIDL that suggests using a transaction processing system message definition to generate an XML document template.

[0010] Blackman teaches building object oriented software to interconnect object oriented applications with non-object oriented datastores such as IMS DBMS from IBM. Blackman teaches persistent object-oriented software to access the IMS data in IMS datastores. Blackman teaches software for accessing IMS data, not interacting (exchanging messages) with an transaction processing application. The software in Blackman is new middleware relating to fields and rows of a database. Blackman does not mention or suggest any teachings regarding

transaction processing system message definitions. In the present invention, transaction processing system message definitions are merged with transaction processing system messages to enable interaction with existing, stable, and reliable, transaction processing applications. In addition, there is no teaching in Blackman that suggests using a transaction processing system message definition to generate an XML document template.

[0011] Transaction processing system message definitions are specific structures that identify an transaction processing application to service a transaction processing system message, and various fields for transaction processing system messages sent to a specific transaction processing application. *See* Specification page 4, line 2 - page 5, line 5. The transaction processing application is identified so that the transaction processing system can schedule processing of a transaction relating to the transaction processing system message. The transaction processing system messages definitions are used to define input and output messaging with the transaction processing application. The transaction processing system message definitions are defined when the transaction processing application is written (source code is created) to enable terminals and other devices to interact with the transaction processing application. Transaction processing system messages may or may not result in the transaction processing application accessing data such as IMS data.

[0012] WIDL teaches a Web Interface Definition Language. The WIDL reference teaches an implementation of XML for defining an interface having services and bindings. *See* WIDL section 5.1. This means that all WIDL files conform to the rules and formatting requirements defined in XML. The WIDL defines an interface that enables non-web-based applications to access web resources. The web resource may be a web page or a functional module. Consequently, the WIDL defines a method that an application can call to get a specific service performed by the web service URL. *See* WIDL section 5.2, service attribute. The method is called by activating the web service URL.

[0013] Claim 1 further recites in pertinent part “and merging a **transaction processing system message** with the generated **template** to produce a corresponding **XML document**.” (emphasis added) The mere mention of IMS in a Blackman which deals with software for enabling object-oriented applications to access IMS data records does not suggest merging of an

transaction processing system message with a template to produce an XML document.

transaction processing system messages are not disclosed, so there is also no enabling disclosure of transaction processing system messages in Blackman.

[0014] Applicant finds no teaching in Blackman or WIDL to generate a template in any format based on I/O message formats, transaction processing system message definitions. The definition of template cited by the Examiner omits one important characteristic of a template. A template leaves openings for data to be inserted. WWW.WEBOPEDIA.COM includes the following definition of template “In spreadsheet and database applications, a template is a blank form that shows which fields exist, their locations, and their length. In spreadsheet applications, for example, a template is a spreadsheet in which **all the cells have been defined but no data** has yet been **entered**.” (Emphasis Added).

[0015] Applicant finds no teaching or suggestion in Blackman or WIDL that certain data has not yet been entered into a structure, a template. Instead, WIDL defines a language which may be used to define a web interface. If a user defines a web interface according to WIDL, there are no defined fields for which data has not yet been entered. Required data attributes defined in the DTD of the WIDL must be defined. In contrast, the present invention claims an XML document template that includes fields, also referred to as placeholders, defined to receive the data from a transaction processing system message once the merging operation occurs. *See* Specification page 7, lines 1-6, and page 26, lines 1-5.

[0016] Applicant asserts that claims 1-5, 11-15, and 21-25 are allowable because Blackman and WIDL fail to teach or suggest use of a transaction processing system message definition to generate an XML document template or merger of a transaction processing system message with the XML document template to produce an XML document. Claims 2-5, 11-15, and 21-25 either depend from claim 1 or recite substantially the same elements. Therefore, Applicant respectfully asserts that claims 2-5, 11-15, and 21-25 are allowable for at least the same reasons as claim 1. Applicant respectfully requests that this rejection be withdrawn.

[0017] In addition, Applicant asserts that the Examiner has failed to establish a *prima facie* case of obviousness because the prior art references neither alone nor in combination teach or suggest all the claim limitations. *See* MPEP § 2142. Specifically, claim 1 recites “generating

an XML document template from **a transaction processing system message definition** and merging **a transaction processing system message** with the generated template to produce a corresponding XML document.” (emphasis added)

[0018] Applicant submits that Blackman and WIDL fail to even mention the use of a transaction processing system message definition or a transaction processing system message, as described above. IMS is mentioned in Blackman but one of skill in the art is given no direction or teaching regarding use of transaction processing system message definitions. Instead, one of skill in the art is taught about record layouts that may be captured from software source code. See Blackman col. 8, lines 38 – 52. These record layouts define the structure and organization of data in the IMS datastores not a transaction processing system message definition. Blackman clearly teaches generating of database descriptions for accessing data, not message definitions that relate to interfacing with a transaction processing system application.

[0019] As amended, transaction processing system message definitions and transaction processing system message processing are specific limitations recited in the independent claims 1, 11, and 21. The plain meaning of these terms requires that any interpretation of the terms be limited to transaction processing system messaging systems, structures, or methods. There is no mention of these terms in Blackman or WIDL.

[0020] Finally, Applicant respectfully asserts that the Examiner has failed to establish a *prima facie* case of obviousness because there is no objective evidence of record to prove a suggestion or motivation to combine the references. *In re Lee*, 277 F.3d 1338, 61 USPQ2d 1430 (Fed. Cir. 2002). Assuming for a moment that the prior art references did teach all the limitations of the claims, “it is insufficient that the prior art disclosed the components of the patented device, either separately or used in other combinations; there must be some teaching, suggestion, or incentive to make the combination made by the inventor.” *Northern Telecom, Inc. v. Datapoint Corp.*, 908 F.2d 931, 934 (Fed. Cir. 1990). See e.g. *Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed.Cir.1985). The Examiner has failed to provide objective evidence of a teaching, suggestion, or incentive to make the combination made by the Applicant.

[0021] The Examiner sets forth the following motivation for combining Blackman and

WIDL: "...because such a combination would allow users of Blackman et al. the benefit of WIDL definitions [which] provide a mapping between Web resources and applications written in conventional programming languages such as..." Applicant fails to see the benefit of the combination, as the combination relates to transaction processing system message processing. As discussed above, Blackman relates to enabling object-oriented applications to access transaction processing system datastores and WIDL refers to a definition language for defining an interface between the object-oriented applications and a web resource such as transaction processing system data.

[0022] Therefore, the benefit suggested by the Examiner is for any application accessing an IMS datastore. The benefit is on the data level between data objects and application objects. In contrast, the present invention involves enabling any application to interface with existing transaction processing system applications on the application level. The present invention facilitates inter-application communication between unmodified, stable, existing transaction processing system applications and modern applications. Applicant respectfully asserts that the benefit suggested by the Examiner on the data level is fundamentally different from the inter-application benefits provided by the present invention. Therefore, because the benefit proposed by the Examiner is not applicable, the Examiner has failed to provide objective evidence of a teaching, suggestion, or incentive to make the combination made by the Applicant.

[0023] The Federal Circuit has recently stated:

[A] rejection cannot be predicated on the mere identification ... of individual components of claimed limitations. Rather, particular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed.

*In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000). Applicant asserts that the Examiner has failed to provide objective evidence from the prior art why one would make the combination suggested by the Examiner. Instead, Applicant respectfully submits that the motivation for the combination has been derived from the Applicant's disclosure.

[0024] Therefore, Applicant asserts that claims 1, 11, and 21 are allowable because the Examiner has failed to establish a *prima facie* case for obviousness. Claims 2-5, 12-15, and 22-25 either depend from independent claims 1, 11, or 21, or recite substantially the same elements.

Therefore, Applicant respectfully asserts that claims 2-5, 12-15, and 22-25 are allowable for at least the same reasons as the independent claims. Applicant respectfully requests that this rejection be withdrawn.

**REJECTION OF CLAIMS 6, 16, AND 26 UNDER 35 U.S.C. §103(a)**

[0025] The Examiner rejected claims 6, 16, and 26 in view of Blackman, WIDL, Lyengar, and Brodsky. Claims 6, 16, and 26 depend from independent claims 1, 11, and 21. Therefore, Applicant respectfully submits that claims 6, 16, and 26 are allowable for at least the same reasons as the independent claims discussed above.

**REJECTION OF CLAIMS 7-10, 17-20, AND 27-30 UNDER 35 U.S.C. §103(a)**

[0026] The Examiner rejected claims 7-10, 17-20, and 27-30 in view of Blackman, WIDL, and Friedman. Claims 7-10, 17-20, and 27-30 depend from independent claims 1, 11, and 21. Therefore, Applicant respectfully submits that claims 7-10, 17-20, and 27-30 are allowable for at least the same reasons as the independent claims.

**AMENDMENTS TO THE CLAIMS**

[0027] In view of the telephone conference held on March 9, 2004 and to advance prosecution, Applicants have made clarifying amendments to the independent claims 1, 11, and 21. Specifically, Applicants have clarified that the message definitions in IMS (a software product from IBM for which IMS is a registered trademark) are definitions used by a particular type of software product, namely a “transaction processing system.” Other amendments were made to clarify that “IMS” refers to a “transaction processing system,” one example of which is the IMS software product from IBM.

[0028] In addition, an amendment clarifies that the message definition sets forth how messages are to be formatted for a transaction and what elements of the message will mean (syntax and semantics). Support for this amendment is found in the specification on page 4, line 11 through page 5, line 5. Applicants further submit that the art of record also fails to teach or disclose message definitions that define transaction messages exchanged with a transaction

processing system, such as IBM's IMS, using XML documents as recited in the independent claims.

[0029] In view of the foregoing, Applicant submits that the application is in condition for immediate allowance. In the event any questions or issues remain that can be resolved with a phone call, the Examiner is respectfully requested to initiate a telephone conference with the undersigned.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "David J. McKenzie", written over a horizontal line.

David J. McKenzie

Reg. No. 46,919

Attorney for Applicant

Date: April 13, 2004

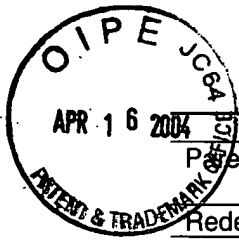
8 East Broadway, Suite 600

Salt Lake City, UT 84111

Telephone (801) 994-4646

Fax (801) 322-1054





Parent ID	f 4	The symbol ID of the immediate parent of the symbol being defined.
Redefined ID	f4	The symbol ID of the data item, that this item renames.
Symbol Name Length	h 2	The number of characters in the symbol name.

## 5.2 Obtaining the needed Information

Both the source code and the SysAdata file contain the same information. In order to read the source code, a parser would need to understand a subset of the COBOL language. Implementing this subset without a language specification by making up own rules will work, but it is almost impossible to cover all possible exceptions that might occur. The Adata file is in a defined format, and every single bit has a defined meaning. Therefore it is easier to read and understand.

A COBOL compiler produces the Adata file when it is able to compile the source file without major errors. It has to be verified that the compilation completed with a return code of 4 or less: analysis can proceed if there are Information and Warning level messages but there must be no Error, Severe error, or Termination messages.

To provide a valid source file to the compiler, the message definition (taken from either the working-storage section of an application program or copy file) definition can be copied into the WORKING-STORAGE-SECTION of a COBOL source file template. The above example for an input message (see page 13) as a valid input file to the compiler looks as follows:

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. EXAMPLE-MSG.  
ENVIRONMENT DIVISION.  
DATA DIVISION.
```

```
WORKING-STORAGE SECTION.  
01 INPUT-MSG.  
02 IN-LL          PICTURE IS 9(2).  
02 IN-ZZ          PICTURE IS 9(4).  
02 IN-TRAN        PICTURE IS X(10).  
02 IN-COMMAND     PICTURE IS X(8).  
02 TEMP-COMMAND REDEFINES IN-COMMAND.  
04 TEMP-IOCMD     PIC X(3).  
04 TEMP-FILLER    PIC X(5).
```

```
PROCEDURE DIVISION.  
STOP RUN.
```

It would also possible to read in a SysAdata file from the compilation process of an entire application. But since one application can define various messages in the working-storage section, the user would need to make a choice which message definition to translate into a XML document. This prototype does not allow the user to make a choice, it translates the entire working-storage section.

The whole process of creating a XML document from a COBOL message definition includes:

1. The message source has to be extracted from a copy file or the source code